Geometry Inspired Algorithms for Linear Programming

Dhananjay P. Mehendale
Sir Parashurambhau College, Tilak Road, Pune-411030,
India

Abstract

In this paper we discuss some novel algorithms for linear programming inspired by geometrical considerations and use simple mathematics related to finding intersections of lines and planes. All these algorithms have a common aim: they all try to approach closer and closer to "centroid" or some "centrally located interior point" for speeding up the process of reaching an optimal solution! Imagine the "line" parallel to vector *C*, where $C^T x$ denotes the objective function to be optimized, and further suppose that this "line" is also passing through the "point" representing optimal solution. The new algorithms that we propose in this paper essentially try to reach at some feasible interior point which is in the close vicinity of this "line", in successive steps. When one will be able to arrive finally at a point belonging to small neighborhood of some point on this "line" then by moving from this point parallel to vector *C* one can reach to the point belonging to the sufficiently small neighborhood of the "point" representing optimal solution.

1. **Introduction:** There are two types of **linear programs** (linear programming problems):

    1. Maximize: $C^T x$
       Subject to: $Ax \le b$
       $x \ge 0$
       Or
    2. Minimize: $C^T x$
       Subject to: $Ax \ge b$
       $x \ge 0$

    Where x is a column vector of size $n \times 1$ of unknowns.
    Where C is a column vector of size $n \times 1$ of profit (for maximization problem) or cost (for minimization problem) coefficients, and C^T is a row vector of size $1 \times n$ obtained by matrix transposition of vector C.
    Where A is a matrix of constraints coefficients of size $m \times n$.

Where b is a column vector of constants of size $m \times 1$ representing the boundaries of constraints.

By introducing the appropriate slack variables (for maximization problem) and surplus variables (for minimization problem), the above mentioned linear programs get converted into **standard form** as:

Maximize: $C^T x$
Subject to: $Ax + s = b$ (1.1)
$x \geq 0, s \geq 0$

Where s is slack variable vector of size $m \times 1$.
Or

Minimize: $C^T x$
Subject to: $Ax - s = b$ (1.2)
$x \geq 0, s \geq 0$

Where s is surplus variable vector of size $m \times 1$.

In the geometrical language, the constraints defined by the inequalities form the so called **convex polyhedron** bounded by the constraint planes, $Ax = b$, and coordinate planes, $x = 0$, and it is straightforward to check that there exists at least one vertex, of this polyhedron at which the optimal solution for the problem is situated when the problem at hand is well defined having at least one solution and not unbounded or infeasible one. Further, we denote the plane defined by equation $C^T x = d$, for some chosen value, d, as the **objective plane**. Many a times there exists be unique optimal solution but sometimes there may exist many optimal solutions, e.g. when one of the constraint planes and the objective plane are parallel to each other then we can have a multitude of optimal solutions.

The method that is proposed in this paper attempts to utilize the geometrical structure of the linear programming problem. As mentioned above, in geometrical language every linear programming problem defines a convex polyhedron formed by intersecting constraint planes and coordinate planes. The region inside of this convex polyhedron is called the **feasible region**. It is made up of **feasible interior points** lying inside this convex polyhedron. These so called feasible interior points satisfy all boundary constraints and are nonnegative. Solving the maximization (minimization) linear programming problem in geometrical terms essentially consists of pushing this objective plane outwards (inwards) in the direction of C such that this objective plane will reach to the extreme end of the polyhedron and will contain the **extreme vertex**. This extreme vertex is actually the **optimal solution** of the problem at hand. In geometrical terms solving a linear programming problem means to arrive at the point representing extreme vertex, to determine coordinates of this point, and further to find value of the objective function (the so called optimal objective value) at this point.

2. **New Algorithms for Linear Programming:** Because of the far great practical importance of the linear programs and other similar problems in the operations research it is most desired thing to have an algorithm which works in a **single step**, if not, in as few steps as possible. No method has so far been found which will yield an optimal solution to a linear program in a single step ([1], Page 19).

   We wish to emphasize in this paper that when one is lucky enough to arrive at some interior feasible point belonging to the "line" parallel to vector C which is also passing through the point representing optimal solution, where $C^T x$ denotes the objective function to be optimized, then one can find the point representing optimal solution to the linear program under consideration in a single step by just moving along this "line" till one reaches the desired extreme vertex on this "line" at the boundary of the convex polyhedron!!

   We wish to develop some NEW algorithms for solving linear programming problems. We now proceed with brief description of these algorithms. We will try to capture the essence of the methods with the help two geometrical figures, FIG.1 and FIG.2 given below. We hope that these figures will make explicit the geometrical motivation that lies behind these algorithms.

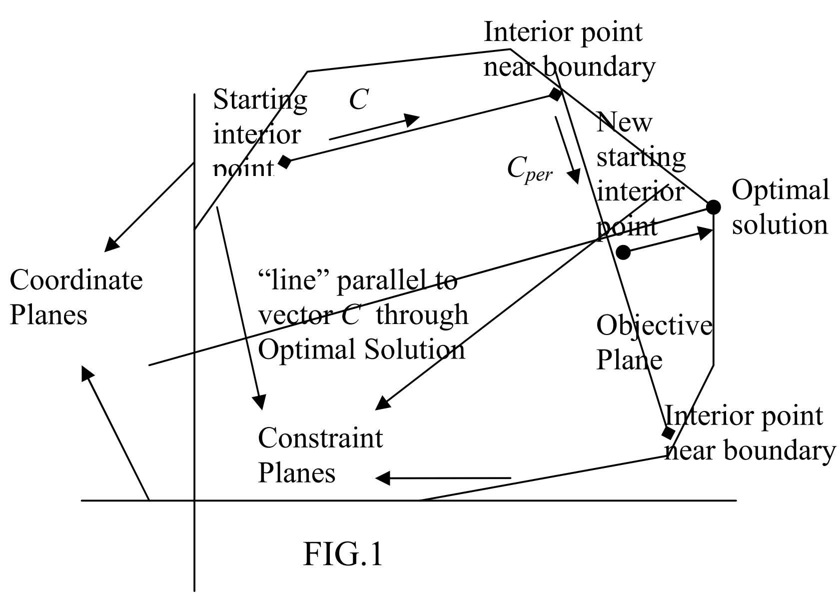

FIG.1

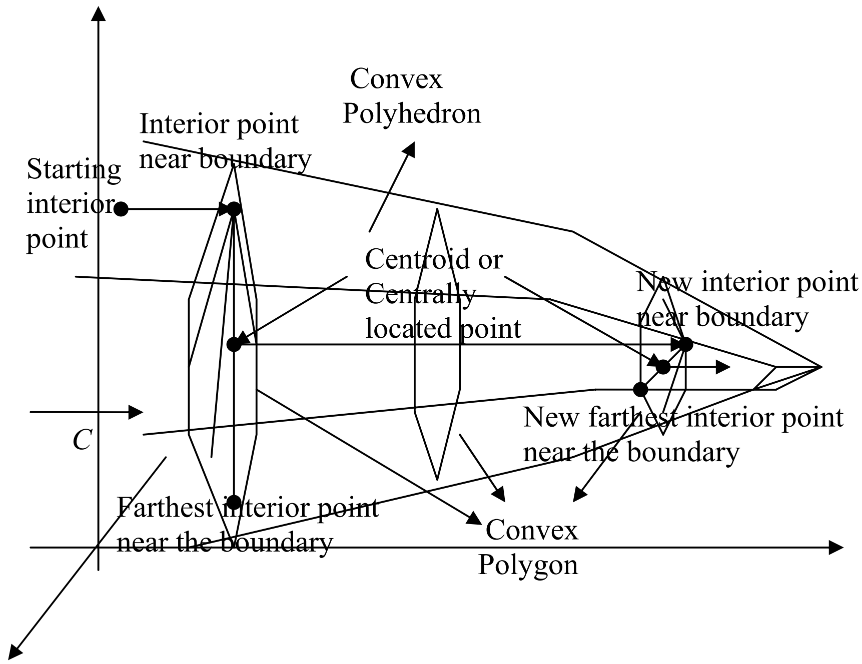

FIG.2

A Brief Description of Figures:

In FIG. 1 we have chosen for the sake of simplicity a typical two dimensional linear programming problem, i.e. a problem which contains two variables, say *x* and *y*. In this 2-dimensional case the objective plane, constraint planes, coordinate planes, are all actually lines as shown in this figure. Vector *C* which is perpendicular to objective plane (a line in this case) is shown. Also, the (the most important) "line" parallel to vector *C* and passing through the point representing optimal solution (extreme vertex) is shown. A vector perpendicular to vector *C*, say C_{per}, is shown. So called starting point, which is some feasible interior point is shown. Starting at this so called "starting" interior point we move in the direction of *C* till we reach at a farthest possible feasible interior point near boundary. In this step the value of objective function improves. Now, from this interior point near boundary we move in the direction of C_{per} till we reach at another interior point near boundary lying in this direction. In this step the value of objective function remains the same, as this new point is on the same objective plane. This step is carried out to locate the desired centrally located point. We take it as midpoint of line segment joining these two interior points near the boundary and take this midpoint as new starting point to start next iteration and to move in the direction of *C* till we again reach at a new interior point near boundary, and so on.

In FIG.2 we have tried to depict higher dimensional case of a linear programming problem, i.e. a problem containing at least three variables (or more: say *n* variables). The constraint planes and coordinate planes give rise to a convex polyhedron (in *n* dimensional space) enclosing feasible interior points as shown in the figure. The objective plane passing through some feasible interior point intersects with the polyhedron and the intersection is a convex polygon as shown in the figure. Vector *C* which is always perpendicular to (any) objective plane as shown. So called starting point, which is some interior point is shown. Starting at this so called "starting" interior point we move in the direction of *C* till we reach at an interior point near boundary. We consider intersection of the objective plane passing through this interior point near boundary with polyhedron which is a polygon as shown. We determine *n* points lying on the boundary (edges of convex polygon). We use these points (in place of vertices) and find the centriod of this polygon. We take this centroid as new starting interior point to start next iteration and move along the line parallel to vector C passing through this centroid till we reach a new farthest possible feasible interior point near boundary, and so on. The other method find a centrally located point as mean of interior point near boundary and farthest interior point near boundary as shown in figure. The farthest interior point is found here using methods of calculus. For this purpose, we maximize the distance function. We consider line segment joining the point we arrived at near the boundary and the other interior point lying on the polygon at **assumed maximum distance**. We then take the mean of these two points as centrally located point to be used as new starting point to start new iteration i.e. to move in the direction of *C* till again we reach at a new interior point near boundary, and so on.

A Brief Description of First Algorithm: The first algorithm starts at some feasible interior point. One then moves along the line passing through this point and parallel to vector *C* in outward (inward) direction for maximization (minimization) problem till one will reach at an interior point very close to the boundary of the convex polyhedron defined by the problem. One then finds the value of objective function (objective value) at this newly obtained feasible interior point. One then considers objective plane passing through this point and so having this newly obtained objective value (for all its points). One then considers the intersection of this objective plane, which has newly obtained objective value for its points, with the convex polyhedron defined by the problem. This intersection will be a convex polygon. One then finds out the centroid of this polygon. This centroid will be the new starting point to move along the line passing through this centroid and parallel to vector *C* in outward (inward) direction for maximization (minimization) problem again till one will reach at an interior point very close to the boundary of the convex polyhedron defined by the problem. One then repeats the same earlier steps of finding the value of objective function (objective value) at this newly obtained feasible interior point, considering the objective plane passing through this point and so having this newly obtained objective value (for all its points), considering the intersection of the objective plane, having newly obtained objective value for its points, with the convex polyhedron defined by the problem, which will again as previous will be a polygon. One then finds out centroid of this new polygon, again moves along the line passing through this centroid and parallel to vector *C* in outward (inward) direction for maximization (minimization) problem again till one will reach at an interior point very close to the boundary of the convex polyhedron defined by the problem. The steps mentioned above taken repeatedly will make us arrive at different objective planes in succession which will have improved value for the objective function, and different polygons will be formed in succession by the intersection of these objective planes with the convex polyhedron defined by the problem. These polygons formed in succession will be having smaller and smaller size (area). In each iterative step one thus will be going in the direction parallel to vector *C* along the line passing through the centroid of these polygons with reduced area generated through intersection of objective planes arrived at in successive steps with the convex polyhedron. It is straightforward to see that in successive steps of this algorithm one will clearly arrive finally either at a point on the “line” or at a point belonging to small neighborhood of some point on the “line” which is parallel to vector *C* and also passing through the “point” representing optimal solution. By moving then from this point on the “line” or point belonging to small neighborhood of a point on the “line” parallel to vector *C* one can reach to the “point” representing optimal solution or to the point belonging to the close neighborhood of the “point” representing optimal solution.

A Brief Description of Second Algorithm: The steps of the second algorithm are almost identical with that of the first algorithm. It also starts at some feasible interior point. One then moves in the direction of vector *C* till one reaches at a feasible interior point lying just inside the boundary of the convex polyhedron defined by the problem in that direction. One then considers the objective plane passing through this point which intersects with the convex polyhedron and again as previous this intersection is a polygon. In the first algorithm we found the centroid of this polygon by finding and using certain *n* points on the boundary of this polygon. In this second algorithm we differ in this step. Instead of finding centroid we find some other point which is also centrally located. Starting with the feasible interior point just obtained in the previous step **one finds the direction** in which if one will proceed along some line lying on the objective plane passing through just obtained feasible interior point near the boundary till one reaches at a feasible interior point on the other side near the boundary of the convex polygon then this feasible interior point will be **at longest possible distance**. By this choice for the **direction** to proceed from obtained feasible interior point near the boundary of the polygon to some other interior point on the other side of the boundary such that the length of this line segment joining these points is largest. By using methods of calculus we manage to maximize the distance between the above mentioned two boundary points so that we make this line segment as longest line segment that resides on the polygon. This obviously makes this line to proceed through central region of the polygon for ensuring largest possible length for this line segment. Thus, by moving in this direction one will be moving in the close vicinity of the centroid! We find midpoint of the line segment joining of these two extreme feasible interior points. This midpoint will automatically be close to centroid as desired. One starts the new iteration here and again one moves in the direction parallel to vector *C* along the line passing through this centrally located midpoint of the segment till one arrives at some new point lying on the boundary of the convex polyhedron in the direction of vector *C*. Thus, almost all the steps of second algorithm are identical to first algorithm except the step of finding centroid. Instead, here one finds a centrally located point (something similar to centroid) which also works well to achieve same results as the first algorithm.

A Brief Description of Third Algorithm: The steps of the third algorithm are again almost identical with that of the earlier algorithms. It differs in its method of locating centrally located point. It also starts at some feasible interior point. One then moves in the direction of vector *C* till one reaches at a feasible interior point lying just inside the boundary. From this point near the boundary we draw perpendiculars on each of the constraint planes (as well as the coordinate planes) and find these foots of perpendiculars. Using these points which are foots of perpendiculars we find centroid and use it as new starting point and move in the direction parallel to vector *C* along the line passing through this centroid till one arrives at some new interior point lying on the boundary, and so on. Instead of dropping perpendiculars on the constraint planes one can drop them on the edges of convex polygon that results through the intersection of objective plane passing through just obtained feasible interior point near boundary intersects with convex polyhedron defined by the problem under consideration. We use thus obtained foots of perpendiculars lying on the edges of the polygon to find centroid and use this centroid as new starting point to start a new iteration and again move in the direction parallel to vector *C* along the line passing through this centroid till one arrives at some new interior point lying on the boundary, and so on.

3. A Detailed Description of Algorithms:

First Algorithm

1) We take some interior point to start with, say x^s,

$$x^s = (x_1^s, x_2^s, \cdots, x_n^s)$$

2) We then move along the line passing through this point, x^s, and parallel to vector *C* in outward (inward) direction for maximization (minimization) problem till one will reach at an interior point very close to the boundary of the convex polyhedron defined by the problem, say x^{i_1}.

3) We then find the objective value (value of objective function) at x^{i_1}, namely,

$$C^T x^{i_1} = d^{i_1}$$

4) We then form the objective plane defined by the equation,

$$C^T x = d^{i_1}$$

This objective plane intersects the convex polyhedron defined by the problem at hand and gives rise to a polygon. We wish to find centroid of this polygon and for this purpose we proceed to find the points, one on each edge of this polygon, say, $x^1, x^2, \cdots, x^m$, as described in the following step.

Note that the matrix equation, $Ax = b$, actually together represent *m* number of equations representing *m* number of constraint planes. These constraint planes together with coordinate planes, $x = 0$, give rise to convex polyhedron.

5) We solve together the equation defining first constraint plane say, $A_1 x = b_1$, where A_1 represents first row of matrix A, and the objective plane $C^T x = d^{i_1}$ and find solution representing the first point (vertex), say x^1. We solve together the equation defining second constraint plane say, $A_2 x = b_2$, where A_2 represents second row of matrix A and $C^T x = d^{i_1}$ and find the solution representing the second point (vertex), say x^2. We continue solving pairs of equations till finally we will solve together the equation defining *m*-th constraint plane say, $A_m x = b_m$ and $C^T x = d^{i_1}$ and find solution representing the *m*-th point (vertex), say x^m.

6) We now find the centroid, namely,

$$R = \frac{\left(\sum_{i=1}^{m} x^i\right)}{m} = x^{ns}$$

This centroid will be used as a new starting point to proceed. We move in the direction of vector C, along the line passing through this centroid till again as previous we reach at an interior point very close to the boundary of the convex polyhedron defined by the problem, say x^{i_2}. We now treat this point x^{i_2} as x^{i_1} (i.e. $x^{i_1} \leftarrow x^{i_2}$) and go to step 3) and begin next iteration of the algorithm.

7) We continue iterations for sufficiently many times, which will produce polygons of smaller and smaller sizes in successive iterations, and this will finally take us to the point which represents optimal solution of the problem, or in very small neighborhood of such a point which represents optimal solution of the problem.

Second Algorithm: This second algorithm is almost identical to first algorithm. All the steps except one are same. The only difference between this algorithm and the first algorithm is in the procedure that we follow here to find a centrally located point on the polygon. We therefore discuss **only this step of finding centrally located point** in order to avoid unnecessary repetition. In the first algorithm we take centroid as such a point. In this algorithm we don't find and take centroid as such a point but obtain the required centrally located point through different considerations. It is straightforward to see from geometry that each time (i.e. in each iterative step) finding and choosing some centrally located point as a starting point to move parallel to vector *C* till one reaches very near to the boundary of polyhedron will take us near the point that represents optimal solution in a much faster way causing **substantial improvement** in the value of the objective function in each step. Choosing some other point on the polygon which is away from a centrally located point to move parallel to vector *C* till one reaches very near to the boundary of polyhedron will show much slower improvement and so will demand large many steps to reach the point which either itself the optimal point or lying in very small neighborhood of optimal point.

As is done in first algorithm we start at some interior point, x^s say and proceed along a line through this point which is parallel to vector *C*, till we reach at an interior point near the boundary of polyhedron, say x^f. Thus, x^f is the farthest interior point on the line through x^s and parallel to vector *C*. One can write

$$x^f = x^s + \alpha C$$

Where scalar α is so chosen that $Ax^f \leq b$ is satisfied. To find such α we put x^f in the first constraint, i.e. we consider

$$A_1 x^f \leq b_1$$

and find condition on α like $\alpha \leq l_1$. We then put x^f in the second constraint, i.e. we consider $A_2 x^f \leq b_2$

and find condition on α like $\alpha \leq l_2$. We continue in this way and consider every constraint in succession and consider finally the *m*-th constraint, i.e. we consider $A_m x^f \leq b_m$

and find condition on α like $\alpha \leq l_m$. We then set

α = min $\{l_i\}$.With this α we get farthest possible interior point in the direction of vector C on the line passing through x^s.

As done in previous algorithm we consider objective plane passing through x^f and consider its intersection with polyhedron which will be a convex polygon. In the first algorithm we obtained centroid of this polygon to treat it as a new starting point. Now, for this algorithm we wish to consider various lines passing through point x^f and lying in the objective plane defined by equation $C^T x = d^f$ having θ as the angle between any two neighboring lines. Let us suppose that $x_1^w, x_2^w, \cdots, x_m^w$ are the feasible interior points lying on these lines through x^f going in different directions and at the farthest possible distance from point x^f, the maximal nature of these distances is ensured by observing the fulfillment of constraints maximally. We then aim to search out the line such that for the obtained point, say x_k^w, among the distances $| x^f - x_i^w |, i = 1,2,\cdots$ the distance $| x^f - x_k^w |$ is maximum. It is easy to check that such line will pass closely to centroid or centrally located point.
As a simplified version for the above procedure let us consider following **different lines** lying on this polygon and passing through the interior point x^f. All these lines will belong to the objective plane through x^f, namely, $C^T x = d^f$, where $d^f = C^T x^f$ and so will be **perpendicular** to vector C. Let us denote a vector belonging to objective plane through x^f, by symbol $C_\perp$ (or C_{per}), then clearly, $C.C_\perp = 0$. Let $C_\perp = (\omega_1, \omega_2, \cdots, \omega_n)$. Using $C.C_\perp = 0$ we can eliminate one parameter and in the expression for $C_\perp$. Thus, we can have $\omega_1 = \phi^1(\omega_2, \omega_3, \cdots, \omega_n)$ and therefore we can write $C_\perp = (\phi^1, \omega_2, \cdots, \omega_n)$ where ϕ^1 is function of parameters

$\omega_2, \omega_3, \cdots, \omega_n$. Similarly, we can have $\omega_2 = \phi^2(\omega_1, \omega_3, \cdots, \omega_n)$ and therefore we can write $C_\perp = (\omega_1, \phi^2, \cdots, \omega_n)$, On continuing on these lines we can have variety of expressions for $C_\perp$,like $C_\perp = (\omega_1, \omega_2, \phi,^3 \cdots, \omega_n)$, ..., $C_\perp = (\omega_1, \omega_2, \cdots, \phi^n)$.

Now **in order to generate various lines** passing through x^f and lying on the objective plane defined by equation $C^T x = d^f$ let us assign unit values to the parameters in functions. We can build in this way different vectors perpendicular to vector C , like $C_\perp^1 = (\phi^1(1,1,\cdots,1),1,\cdots,1)$, $C_\perp^2 = (1, \phi^2(1,1,\cdots,1),\cdots,1)$, ..., $C_\perp^n = (1,1,\cdots,\phi^n(1,1,\cdots,1))$. Now, from the above obtained vectors perpendicular to C we build following vectors like , $x^{g_1} = x^f + \mu_1 C_\perp^1$, $x^{g_2} = x^f + \mu_2 C_\perp^2$, $x^{g_3} = x^f + \mu_3 C_\perp^3$, ……, $x^{g_n} = x^f + \mu_n C_\perp^n$. For each vector we find the largest possible value for μ_i such that all the constraints remain valid and the vectors x^{g_i} represent feasible interior points. Among these vectors x^{g_i} we choose the one for which the Euclidean distance between the chosen vector say, x^{g_k} and vector x^f is maximum. It is easy to check that for such vector x^{g_k} the line joining the points represented by vectors x^{g_k} and x^f will pass nearest to a centrally located point among these vectors x^{g_i} . We now choose a **new starting point** to move in the direction of vector C, namely, $x^{ns} = \frac{1}{2}\left(x^f + x^{g_k}\right)$. It is easy to check that all these steps will finally take us to the point which represents optimal solution of the problem, or in very small neighborhood of such a point which represents optimal solution of the problem.

In the method just discussed to locate a centrally located point we have suggested to test and find from several vectors $x^{g_i} = x^f + \mu_i C_\perp^i$ the **best one**. We now proceed to see a **smarter method** to find a centrally located point. In this method we make use of the **methods of calculus** as follows: We consider a **general point** represented by vector x^g on the objective plane passing through point represented by vector x^f, thus $x^g = x^f + \mu C_\perp$, whose coordinates are *n* parameters. We correlate and determine these parameters using **maximal nature of distance** between points represented by points x^f and x^g. When these points x^f and x^g are maximally separated we can expect these points x^f and x^g to be situated diametrically opposite to each other with respect to a circle enclosing the polygon. So, further we can safely expect that the line joining points represented by respectively x^f and x^g will pass closely to centrally located point (of the polygon) if the Euclidean distance between points represented by respectively x^f and x^g will be maximum, i.e. $| x^g - x^f | = | \mu C_\perp |$ is maximum. We take $C_\perp = (\phi^1, \omega_2, \cdots, \omega_n)$, setup equations $\frac{\partial(| C_\perp |)}{\partial(\omega_i)} = 0$ for all $i = 2,3,\cdots,n$ and find interrelations between parameters ω_i. As a consequence, we finally get the equation, $x^g = x^f + \mu C_\perp$, where $C_\perp = (\beta_1, \beta_2, \cdots, \beta_n)$ and these β_i are certain constants, and we have to determine largest possible value for μ again using constraints (as it was done above to find largest possible value for α) so that x^g will be farthest interior point from x^f. As is done early, we now find a **new starting point** to move in the direction of vector C, namely, $x^{ns} = \frac{1}{2}\left(x^f + x^g\right)$ It is easy to check that all these steps will finally take us to the point which represents optimal solution of the problem, or in very small neighborhood of such a point which represents optimal solution of the problem. We have thus discussed only that portion of second algorithm that differs and for the entire rest portion both algorithms proceed in identical way!

<u>Third Algorithm:</u> This third algorithm is almost identical to first two algorithms. All the steps except one are same. The only difference between this algorithm and the first two algorithms is in the procedure that we follow here to find a centrally located point. We therefore discuss **only this step of finding centrally located point** in order to avoid unnecessary repetition. As is done in previous algorithms we start at some interior point, x^s say and proceed along a line through this point which is parallel to vector *C*, till we reach at an interior point near the boundary of polyhedron, say x^f. Thus, x^f is the interior point near boundary on the line parallel to vector *C* through x^s. From this point x^f we draw perpendiculars on constraint planes, $A_i x = b_i$ for all *i* = 1, 2,.., *m*. and find each foot of perpendicular, $x_1^u, x_2^u, \cdots, x_m^u$. Again as previous we find the centroid, namely,

$$R = \frac{\left(\sum_{i=1}^{m} x_i^u\right)}{m} = x^{ns}$$

This centroid will be used as a new starting point to proceed along the line passing through it in the direction parallel to vector *C*, and so on. As an alternative, we draw perpendiculars on edges of convex polygon, i.e. on lines formed by intersection of each constraint plane, $A_i x = b_i$ for all *i* = 1, 2,.., *m*, with the objective plane, $C^T x = d^f$, where $d^f = C^T x^f$ and find each such foot of perpendicular, $x_1^v, x_2^v, \cdots, x_m^v$. Again as previous we find the centroid, namely,

$$R = \frac{\left(\sum_{i=1}^{m} x_i^v\right)}{m} = x^{ns}$$

This centroid will be used as a new starting point to proceed along the line passing through it in the direction parallel to vector *C*, and so on.

Example 1: Maximize: $x + y$

Subject to: $x + 2y \le 4$

$$-x + y \le 1$$

$$4x + 2y \le 12$$

$$x, y \ge 0$$

Solution: It is easy to check that (1, 1) is a feasible interior point. The value of objective function at this point is

$$x + y = 1 + 1 = 2.$$

So, we start by taking (1, 1) as starting interior point. We now have to proceed in the direction of vector *C* till we reach near boundary and thus reach at an interior point near boundary. Now, *C* = (1, 1) so our new interior point will be

$$(1, 1) + \alpha C = (1, 1) + \alpha\,(1, 1) = (\alpha + 1,\ \alpha + 1)$$

Now, we determine α such that we get interior point near boundary, i.e. a point near boundary such that all the constraints will be satisfied. Substituting the point in the first constraint

$$\alpha + 1 + 2\alpha + 2 \le 4, \text{ i.e. } \alpha \le 1/3$$

Substituting the point in the second constraint

$$-\alpha - 1 + \alpha + 1 \le 1, \text{ i.e. } 0 \le 1$$

Substituting the point in the third constraint

$$4\alpha + 4 + 2\alpha + 2 \le 12, \text{ i.e. } \alpha \le 1$$

Thus, we take $\alpha = \min\{\alpha\} = 1/3$. Using this $\alpha = 1/3$ we get the desired point near boundary, namely,

$$(1, 1) + \alpha C = (1, 1) + \alpha\,(1, 1) = (\alpha + 1,\ \alpha + 1) = (4/3, 4/3)$$

The value of objective function at this point is

$$x + y = 4/3 + 4/3 = 8/3 = 2.66$$

Thus, the value of objective function got improved as expected. We have *C* = (1, 1), therefore, $C_{per} = C_{\perp} = (-1, 1)$. Starting from the just obtained interior point near boundary we now move in the direction of $C_{\perp}$ till we reach the (farthest) interior point on the (other side of the) boundary. Thus, we find

$$(4/3, 4/3) + \beta C_{\perp} = (4/3, 4/3) + \beta\,(-1, 1) = (4/3 - \beta,\ 4/3 + \beta)$$

Substituting this point in the constraint equations we find $\beta \ge -2$ and using $\beta = -2$ we get $(4/3, 4/3) + \beta C_{\perp} = (10/3, -2/3)$, and we then have the centroid as 1/2[(4/3, 4/3) + (10/3, -2/3)] = (7/3, 1/3). Substituting point $(7/3, 1/3) + \alpha C = (7/3, 1/3) + \alpha\,(1, 1)$ in the constraints we find $\alpha = 1/3$. This yields new point near boundary as $(7/3, 1/3) + \alpha\,(1, 1) = (7/3, 1/3) + (1/3, 1/3) = (8/3, 2/3)$. The value of objective function at this point is

$$x + y = 8/3 + 2/3 = 10/3 = 3.3333.$$

Thus, the value of objective function got improved as expected. Actually, one can easily check by carrying out one more iteration that we have already reached the optimal solution!

Example 2: Maximize: $10x_1 + 6x_2 + 4x_3$

Subject to: $x_1 + x_2 + x_3 \le 100$

$$10x_1 + 4x_2 + 5x_3 \le 600$$

$$2x_1 + 2x_2 + 6x_3 \le 300$$

$$x_1, x_2, x_3 \ge 0$$

Solution 1: We have C = (10, 6, 4). Let us take as starting interior point, x^s = (1, 1, 1). Value of objective function at x^s = 20. We now move in the direction of vector C till we reach near boundary and thus reach at an interior point near boundary. So our new interior point will be

$(1, 1, 1) + \alpha C = (1, 1, 1) + \alpha\,(10, 6, 4) = (10\alpha + 1, 6\alpha + 1, 4\alpha + 1)$

Now, we determine α such that we get interior point near boundary, i.e. a point near boundary such that all the constraints will be maximally satisfied. Substituting the point successively in the constraints we get α = min {4.85, 4.034, 5.17} = 4. Thus, we get the point near boundary as (41, 25, 17). Value of objective function at this point is (410 + 150 + 68) = 628. Thus, the value of objective function got improved as expected. We now find point on the intersection of objective plane $10x_1 + 6x_2 + 4x_3 = 628$ separately with each constraint plane and find the intersection points as follows: x^1 = (38, 0, 62), x^2 = (54, 14, 0), x^3 = (49.38, 0, 33.5), therefore, centroid = (47.12, 4.6667, 31.84). Moving in the direction of vector C through centroid we reach to (47.12, 4.6667, 31.84) + 0.1786 (10, 6, 4). Value of objective function at this point is = 653.72. Again, we now find point on the intersection of objective plane $10x_1 + 6x_2 + 4x_3 = 653.72$ separately with each constraint plane and find the intersection points as follows: x^1 = (42.28, 0, 57.7), x^2 = (-100, 276, 0), x^3 = (67.73, 0, -5), therefore, centroid = (3.336, 92, 17.56). Moving in the direction of vector C through centroid we reach to (3.336, 92, 17.56) + 0.1(10, 6, 4). Value of objective function at this point is = 670.9. We thus see that the value of objective function is improving in the successive steps and by continuing these steps we can find out the desired optimal solution.

Solution 2: We have $C = (10, 6, 4)$. Let $C_\perp = (\lambda, \beta, \gamma)$. Using $C.C_\perp = 0$, we have $C_\perp = (-\frac{6}{10}\beta - \frac{4}{10}\gamma, \beta, \gamma)$. We maximize distance, i.e. $F = (-\frac{6}{10}\beta - \frac{4}{10}\gamma)^2 + \beta^2 + \gamma^2$ is maximum. By equating partial derivatives to zero i.e. setting $\frac{\partial F}{\partial \beta} = 0, \frac{\partial F}{\partial \gamma} = 0$ we get $\gamma = -11.3\beta$ or $\gamma = -0.103\beta$. Let us take as starting interior point, $x^s = (1, 1, 1)$. Value of objective function at $x^s = 20$. We now move in the direction of vector C till we reach near boundary and thus reach at an interior point near boundary. So our new interior point will be
$(1, 1, 1) + \alpha C = (1, 1, 1) + \alpha\,(10, 6, 4) = (10\alpha + 1, 6\alpha + 1, 4\alpha + 1)$
Now, we determine α such that we get interior point near boundary, i.e. a point near boundary such that all the constraints will be maximally satisfied. Substituting the point successively in the constraints we get $\alpha = \min\{4.85, 4.034, 5.17\} = 4$. Thus, we get the point near boundary as (41, 25, 17). Value of objective function at this point is (410 + 150 + 68) = 628. Using relation $\gamma = -11.3\beta$ for maximization of distance we have other point $(41, 25, 17) + \mu(3.92, 1, -11.33)$, where using constraints we get $\mu = -1.35$ and so, we have other point = (35.708, 23.65, 32.29). Therefore, the centrally located point = (38.354, 24.325, 24.645). Thus, we consider $(38.354, 24.325, 24.645) + \alpha\,(10, 6, 4)$ and find α using constraints as $\alpha = 0.64$ giving rise to point (44.75, 28.165, 27.205). Thus, value of objective function at this point = 447.5 +168.96 + 98.58 = 715.04. A marked improvement!! We thus see that the value of objective function is improving in the successive steps and by continuing these steps we can find out the desired optimal solution.

Acknowledgements

The author is thankful to Dr. M. R. Modak for some useful discussions.

References

1. Hadley G., Linear Programming, Narosa Publishing House, New Delhi 110 017, Eighth Reprint, 1997.